Learning to Reason with a Restricted View

Learning to Reason with a Restricted View

Roni Khardon
and
Dan Roth

TR-08-96

September 1996

Center for Research in Computing Technology
Harvard University
Cambridge, Massachusetts

Learning to Reason with a Restricted View*

Roni Khardon† **Dan Roth**‡

Aiken Computation Laboratory,
Harvard University,
Cambridge, MA 02138.
{roni,danr}@das.harvard.edu

September 10, 1996

Abstract

The current emphasis of the research in learning theory is on the study of inductive learning (from examples) of concepts (binary classifications of examples). The work in AI identifies other tasks, such as reasoning, as essential for intelligent agents, but those are not supported by the current learning models. The Learning to Reason framework was devised to reconcile inductive learning and efficient reasoning. The framework highlights the fact that new learning questions arise when *learning in order to reason*. This paper addresses the task of deductive reasoning, and investigates learning to reason problems in which the examples seen are only partially specified.

The paper presents several interpretations for partial information in the interface with the environment, and develops model based representations and reasoning algorithms that are suitable to deal with partially observable worlds. Then, learning to reason algorithms that cope with partial information are developed. These results exhibit a tradeoff between learnability, the strength of the oracles used in the interface and the expressiveness of the queries asked.

This work shows that one can learn to reason with respect to expressive worlds, that cannot be learned efficiently in the traditional learning framework and do not support efficient reasoning in the traditional reasoning framework.

1 Introduction

The study of learning is motivated by the belief that a learning component must have a central role in any system capable of performing high level cognitive tasks. The current emphasis of

*An earlier version of the paper appears in the Proceedings of the Workshop on Computational Learning Theory, COLT-95.

†Research supported by ARO grant DAAL03-92-G-0115, by NSF grant CCR-95-04436 and by ONR grant N00014-96-1-0550.

‡Research supported by NSF grant CCR-92-00884, by DARPA AFOSR-F4962-92-J-0466 and by ONR grant N00014-96-1-0550.

the research in learning theory is on the study of inductive learning (from examples) of concepts (binary classifications of examples). In this framework the performance of the learner is measured when classifying future examples and we therefore call this the *Learning to Classify* approach. The theoretical research in this direction, originating from the seminal work of Valiant (1984) has already proved useful in that it had contributed to our understanding of some of the main characteristics of the learning phenomenon and has contributed to applied research on classification tasks (Druker, Schapire, and Simard, 1993).

However, the goal of developing an understanding of how learning supports other high level cognitive tasks is still not within reach. Several such tasks, including language understanding, high level vision and planning, have been widely interpreted as tasks that rely on performing some sort of *inference*. A basic inference task considered in this context is that of *deductive inference*, which is usually modeled as follows: given a propositional expression W, represented as a conjunction of rules and assumed to capture our knowledge of the world, and a propositional query α that is supposed to capture the situation at hand, decide whether W logically implies α (denoted $W \models \alpha$). Early theories of intelligent systems have assumed that these inference tasks can be studied separately from learning. However, in the current frameworks, learning and reasoning algorithms cannot be combined in a straightforward way. For example, even if we could *learn* the set of rules W, we cannot use the outcome to efficiently perform deductive reasoning with the query α. This task requires solving satisfiability for $W \wedge \overline{\alpha}$, a task that is believed to be computationally intractable.

The Learning to Reason framework was introduced in (Khardon and Roth, 1994a) to address the abovementioned problems. On one hand, inspired by the pac-model of learning, it argues that a central question to consider is how the intelligent system acquires its knowledge and how this process, of interaction with its environment, influences the performance of the reasoning system. On the other hand, it highlights the fact that new learning questions arise, and should be addressed if we want to *learn in order to reason.* The Learning to Reason approach combines the interfaces to the world used by known learning models with the reasoning task and a performance criterion suitable for it. In this framework the intelligent agent is given access to its favorite learning interface, and is also given a grace period in which it can interact with this interface and construct a representation KB of the world W. The reasoning performance is measured only after this period, when the agent is presented with queries α from some function class, relevant to the world, and has to answer whether W implies α. It is shown in (Khardon and Roth, 1994a) that a learning algorithm that is aimed at providing good classification behavior does not, in general, facilitate reasoning tasks. At the same time it is shown that in the Learning to Reason approach, through interaction with the world, the agent truly gains additional reasoning power over what is possible in the traditional setting. In particular, cases are presented where learning to reason about the world is feasible but either reasoning from a given representation of the world or learning representations of the world do not have efficient solutions.

In this paper we extend the Learning to Reason framework to study the case in which the interaction with the environment is via partial observations. These interactions are more realistic, but received little attention in learning theory mainly since they become more important when learning in order to reason. (Valiant's (1984) paper and several recent papers discussed in Section 3 are notable exceptions.) When learning to reason one cannot assume that examples include an assignment of values to *all* the attributes in the world, as is reasonable when learning to classify, in tasks such as character recognition. Rather, the information perceived provides only partial information on the state of the world. For example, when sitting in a windowless lecture hall one's senses do not supply any information about the weather. Clearly, some, but possibly not all of the missing information may be relevant to the task at hand.

The agent interacts with the world when learning its environment and when responding to queries. In the latter stage, the interaction is partial in the sense that a query usually specifies and inquires about a small fraction of the world's attributes, inducing a natural way to restrict the type of queries considered. In this paper we concentrate on the interaction in the learning phase. In this stage the learner is exposed to partial information in the form of examples which are not totally specified. We discuss a few ways in which partial examples can be interpreted, including cases where unobserved attributes are irrelevant, and cases where unobserved attributes are arbitrarily (and possibly adversarially) hidden from the learner. The goal of the learner is to reason about this partially observable world. Namely, to answer (deductive) reasoning questions about the world it is interacting with. As in the original work on Learning to Reason (Khardon and Roth, 1994a), it is shown here that Learning to Reason can be achieved even in cases where the traditionally phrased reasoning problem is intractable, and when the traditionally phrased learning problem – learning a representation of the world (a concept) – is intractable.

Our positive results use a model based approach to reasoning (Kautz, Kearns, and Selman, 1995; Khardon and Roth, 1994b). In this approach, the agent keeps a collection of (partial) examples as a knowledge base, and the reasoning task is performed simply by evaluating the queries on this set of models. The need to study these knowledge representations arises from the intractability of reasoning with the standard formula-based representations. We show that such knowledge bases are useful for the deduction task considered here and moreover, that they can be learned, thus producing a system that learns in order to reason. We also prove an impossibility result on learning to reason, for cases in which the class of queries asked are too expressive, and discuss the tradeoffs between the strength of the oracles used and the positive results that can be shown.

While in this paper we concentrate on deductive reasoning, the learning to reason approach should be seen in a more general context and can be applied for a variety of tasks. In particular, a different treatment of partial information is taken in (Valiant, 1995; Roth, 1995), where the effect of partially specified queries is discussed. A similar learning to reason approach is developed there, supporting several aspects of "non-monotonic reasoning" (Reiter, 1987) which have proved difficult to capture in other frameworks. The Learning to Reason approach has also been extended to consider several other tasks, including some forms of default reasoning, learning in order to act in the world, and learning of active classifiers (Khardon and Roth, 1995; Khardon, 1996; Greiner, Grove, and Roth, 1996).

The rest of the paper is organized as follows. We start by presenting some preliminaries on reasoning in Section 2 and then introduce our notion of partial observations in Section 3. In Section 4 we formally define the Learning to Reason model. Section 5 develops ideas on knowledge representations and reasoning procedures required to cope with partially observable worlds. Section 6 presents the Learning to Reason algorithms. We conclude in Section 7 with a summary.

2 Reasoning

The generally accepted framework for the study of reasoning in intelligent systems is the knowledge-based system approach (McCarthy, 1958; Nilsson, 1991). It is assumed that the knowledge is given to the system, stored in some *representation language* with a well defined meaning assigned to its sentences. The sentences are stored in a Knowledge Base (KB) which is combined with a reasoning mechanism, used to determine what can be inferred from the sentences in the KB. There are many knowledge representations that can be used to represent the knowledge in a knowledge-based system. Different representation systems (e.g., a set of logical rules, a probabilistic network) are associated with corresponding reasoning mechanisms, each with its own merits and range of applications. The most basic reasoning task considered is that of *logical deduction*. In this case, reasoning is abstracted as the task of determining whether a *query* α, assumed to capture the situation at hand, is implied from our knowledge of the world as expressed in KB (denoted KB $\models \alpha$). Traditionally, mainly for comprehensibility reasons, the knowledge representation used in these tasks is that of a conjunction of rules, namely a CNF representation.

However, computational considerations render the traditional reasoning approach as well as other variants of it not adequate for common-sense reasoning. This is true not only for the task of deduction, but also for many other forms of reasoning which have been developed, partly in order to avoid the computational difficulties in exact deduction and partly to meet some (psychological and other) plausibility requirements. All those were shown to be even harder to compute than the original formulation (Selman, 1990; Papadimitriou, 1991; Roth, 1996). As a consequence, a lot of recent work in reasoning aims at identifying classes of limited expressiveness, with which one can perform some sort of reasoning efficiently (Levesque and Brachman, 1985; Cadoli, 1995; Levesque, 1992; Selman, 1990). However, none of these works meets the strong tractability requirements for common-sense reasoning (as described, for example, in (Shastri, 1993)), even though, (as argued, for example, in (Doyle and Patil, 1991)) the representation is sometimes restricted in implausible ways.

Moreover, the question of how the knowledge base might be acquired and whether this should influence how the performance of the reasoning system is measured is normally not considered. While the central role of learning in cognition is acknowledged by many, most lines of research nevertheless study reasoning phenomena separately from learning phenomena. The assumption is that "Learning can be added later". That is, the separate study of reasoning and learning phenomena will eventually be combined to produce intelligent behavior (see discussion in (Kirsh, 1991)). Indeed, very few works have considered the question of integrating theories of reasoning and learning in any formal way. As pointed out earlier, computational considerations show that learning and reasoning algorithms cannot be combined in a straightforward way. Other examples for this phenomenon, which do not rely directly on the computational hardness of reasoning, can also be given (Khardon and Roth, 1994a).

Motivated by the above observations, the Learning to Reason framework, suggesting a way to integrate the learning and reasoning tasks, has been developed (Khardon and Roth, 1994a). This is extended in this work to the study of Learning to Reason in the presence of partial observations in the learning stage.

2.1 Formal Definition of the Reasoning Task

We consider problems of reasoning where the "world" (the domain in question) is modeled as a Boolean function[1] $W : \{0,1\}^n \rightarrow \{0,1\}$. Similarly, the knowledge base KB consists of some representation for a Boolean function. Note that we make a distinction between KB and W: KB is the representation used by the algorithm, whereas the reasoning performance is measured relative to W

Let f, g be any Boolean functions. An assignment $x \in \{0,1\}^n$ is a *model* (satisfying assignment) of f if $f(x) = 1$. By "f entails g", denoted $f \models g$, we mean that every model of f is also a model of g. We also refer to the connective $\models$ by its equivalent, proof theoretic name, "implies". Since

[1]This is equivalent to the definition in terms of propositional expressions. A propositional expression is just a representation for a Boolean function, and a propositional language is a class of representations for Boolean functions. These terms are used in the reasoning and learning literature respectively, and we use them interchangeably.

"entailment" and "logical implication" are equivalent, we can treat f either as a Boolean function (usually, using a propositional expression that represents the function), or as the set of its models, namely $f^{-1}(1)$. Observe that the connective "implies" ($\models$) used between Boolean functions is equivalent to the connective "subset or equal" ($\subseteq$) used for subsets of $\{0,1\}^n$. That is, $f \models g$ if and only if $f \subseteq g$. Let $\mathcal{F}, \mathcal{Q}$ be two propositional languages.

Definition 2.1 *An algorithm A is an* efficient and exact reasoning *algorithm for the reasoning problem $(\mathcal{F}, \mathcal{Q})$, if for all $f \in \mathcal{F}$ and $\alpha \in \mathcal{Q}$, when A is presented with input (f, α), A runs in time polynomial in n and the size of f and α, and answers "Yes" if and only if $f \models \alpha$.*

Answering the question $f \models \alpha$ is equivalent to solving unsatisfiability for the formula $f \wedge \overline{\alpha}$. Thus, when f is given as a CNF, exact reasoning can be done efficiently only when satisfiability can be solved efficiently (e.g., Horn expressions). We note that, a DNF representation for f, although better on computational grounds, has been less favored mainly for comprehensibility reasons. (Since a representation as a set of rules easily translates to a CNF but not to a DNF expression.)

The assertion $f \models \alpha$ means that if some model $x \in \{0,1\}^n$ satisfies f, then it must also satisfy α. This motivates the following *model based approach* to solving the deduction problem. Let $\Gamma \subseteq f \subseteq \{0,1\}^n$ be a set of models. To decide whether $f \models \alpha$, check, for all the models $z \in \Gamma$, whether $\alpha(z) = 1$. If for some z, $\alpha(z) = 0$, say "no"; otherwise say "yes".

By definition, if $\Gamma = f$ this approach yields correct deduction, but representing f by explicitly holding *all* the possible models of KB is not plausible. As shown in (Kautz, Kearns, and Selman, 1995; Khardon and Roth, 1994b), under some restrictions, it is sufficient to consider only a small set of models. Our results use this approach to reasoning, and we discuss this issue further in Section 5.

3 Partial Observations

Consider an agent who is wandering around the world gathering information about its environment. We assume the agent interacts with the world via a set of measurements it can make and some predicates it can compute from these measurements. We think of these measurements and predicates as a set of binary attributes and consider a set $X = \{x_1, \ldots, x_n\}$ of variables, each of which is associated with a world's attribute and can take the values 1 or 0 to indicate whether the associated attribute is true or false in the world. While at each point in time the agent can take measurements, in general it would not be possible to observe all the attributes at all times. Thus, while some attributes may be known to be true or false others may not be observed. We denote this situation by assigning the value $*$ to such attributes, so that the input an agent sees is a *partial assignment* to the n variables – an assignment in $\{0,1,*\}^n$. For example, $v = (1 * 0)$ means that x_1 is true, x_3 is false, and the value of x_2 is unknown. An assignment is *total* if the value of every variable is *known* (i.e., assigned value from $\{0,1\}$). An assignment y is an *extension* of x if y agrees with x on all the variables assigned 0 or 1 in x (and where variables assigned $*$ in x may be assigned 0 or 1 in y).

We start by discussing several interpretations for the information conveyed by partial observations. Recall that we model the world as a Boolean function $W : \{0,1\}^n \to \{0,1\}$, where the intention is that $W(x) = 1$ if and only if x corresponds to a combination of features which is possible in the world. There are various ways to interpret the meaning that an observation $v \in \{0,1,*\}^n$ conveys on W. In particular:

1. **Universal interpretation:** For all possible extensions of v to total models v', $W(v') = 1$.

2. **Existential interpretation:** There exists an extension of v to a total model v', such that $W(v') = 1$.

3. **Abbreviated interpretation:** The partial model v is just a short way to write the total model v' defined by $v'_i = v_i$, for all i such that $v_i = 1$, and $v'_i = 0$, otherwise. Thus $W(v) = 1$ means that $W(v') = 1$, for the v' defined above.

In the following we assume that one of the above interpretations has been chosen, and is fixed for the entire duration of the learning scenario. In such a case we can evaluate a Boolean function on a partial assignment according to this interpretation. We denote this by using a subscript to mark the interpretation chosen. Namely, f_u (f_e, f_a) means that the function f is evaluated according to the Universal (Existential, Abbreviated) interpretation.

Several works have dealt with partial observations in the context of concept learning. The approach (1) is taken by Valiant (1984) to model an agent that observes all the attributes that are relevant for the classification of the learned concept. (That is, when one is learning about chairs, the color of the sky at that moment is irrelevant.)

The approach (3) is useful when the total number of attributes n is much larger than the number of positive attributes any one example has, and an example is presented as a list of its positive attributes. Concept learning in this model is studied in (Blum, 1992; Blum, Hellerstein, and Littlestone, 1991).

For the task of learning a "world" representation in order to reason about it later, it seems that the agnostic approach, the existential interpretation (2), ought to be taken. A motivating scenario is that of an agent who is wandering around in the world, but can perceive at any instance only a limited number of attributes. The agent has no control on the perceived attributes, nor can it tell if all the "important" attributes have been perceived. This situation can be modeled by having someone "hide" some of the attributes in an example. As we show later even with this agnostic interpretation some positive results can be achieved.

Several other works have studied partial assignments in the context of learning to classify but used different assumptions on the interface of the agent with the environment. In (Ben-David and Dichterman, 1993) it is assumed that the agent can select the attributes it perceives. In (Hancock, Greiner, and Rao, 1994; Greiner, Grove, and Kogan, 1996) it is assumed that a helpful teacher blocks all the attributes which are irrelevant for classifying an example, thus making the learning task easier. Since the interface and the task considered in both models are different from ours, the results are not directly comparable. Naturally, however, these learning results can be used for Learning to Reason tasks when the output representation supports efficient reasoning.

Schuurmans and Greiner (1994) consider a consistent "blocking process" that hides some of the attributes of a randomly drawn example. In this way their interface is "existential" but makes more assumptions on the missing variables. This approach is also close to ours in the sense that the learning task is to learn "default rules", for the purpose of reasoning with them later. The reasoning stage, however, is not considered, and presumably is performed by a traditional reasoner, and is thus intractable. In recent work Valiant (1995) and Roth (1995) treat the unobserved value $*$ as a third value, and no relation to the true value of the attribute is assumed. They show that several non-monotonic reasoning phenomena can be explained through learning when using this approach.

3.1 Reasoning

Our definition of the deduction task is based on the model-theoretic definition of the implication relation $\models$. That is, $f \models \alpha$ if and only if every model of f is also a model of α. Since the notion of a total model is different than that of a partial model, we need to study the semantics of the implication relation with respect to partial assignments.

The definition of a model can be extended to $\{0,1,*\}^n$, using the interpretations given above. A partial assignment $x \in \{0,1,*\}^n$ is a *p-model* of W if and only if $W_p(x) = 1$, for $p \in \{a, e, u\}$, depending on the interpretation we favor. Likewise we define implication with respect to partial assignments: Let α be a Boolean function. We say that W *p*-implies α ($W \models_p \alpha$) if every p-model of W is also a p-model of α (where $p \in \{a, e, u\}$). As the following theorem shows the connectives $\models$ and $\models_p$ are equivalent, and the semantics so defined preserve the definitions for total models. We therefore use $\models$ in the rest of the paper.

Theorem 3.1 *Let W, α be Boolean functions and $p \in \{a, e, u\}$. Then, $W \models \alpha$ if and only if $W \models_p \alpha$.*

Proof: Assume first that $W \models \alpha$. Let $x \in \{0,1,*\}^n$ such that $W_e(x) = 1$. Then, there exists an extension $x' \in \{0,1\}^n$ of x such that $W(x') = 1$. From the assumption, $\alpha(x') = 1$ and therefore $\alpha_e(x) = 1$. Similarly, let $x \in \{0,1,*\}^n$ such that $W_u(x) = 1$. Then, for all extensions $x' \in \{0,1\}^n$ of x, $W(x') = 1$. Therefore, all those extensions satisfy $\alpha(x') = 1$ and we have that $\alpha_u(x) = 1$. Finally, let $x \in \{0,1,*\}^n$ such that $W_a(x) = 1$. Then, the unique 0-padded extension $x' \in \{0,1\}^n$ of x satisfies $W(x') = 1$. From the assumption, $\alpha(x') = 1$ and therefore $\alpha_a(x) = 1$. We have shown that $W \models_p \alpha$ for all $p \in \{a, e, u\}$.

For the other direction, assume first that $W \models_e \alpha$. Then, given $x \in \{0,1\}^n$ such that $W(x) = 1$, we can treat x as an element of $\{0,1,*\}^n$ and deduce, from the assumption, that $\alpha(x) = 1$. Therefore, $W \models \alpha$. The same argument holds when we assume that $W \models_u \alpha$. Assume now that $W \models_a \alpha$. Then, given $x \in \{0,1\}^n$ such that $W(x) = 1$, we define $x' \in \{0,1,*\}^n$ by replacing all the 0 entries in x by $*$'s. By definition, $W_a(x') = 1$ and therefore $\alpha_a(x') = 1$, and we get that $\alpha(x) = 1$, that is, $W \models \alpha$. ■

We note that our approach is different from several works which have studied non-standard models in the context of reasoning (Levesque, 1984; Fagin, Halpern, and Vardi, 1995). There, in order to solve the so-called logical omniscience problem they allow a single model to assign both 0 and 1 to a variable, thus changing the semantics. As we have shown above, our definition maintains the usual semantics.

3.2 Computational Considerations

While the evaluation of Boolean functions is trivial for total models, model evaluation may be hard for partial assignments. To facilitate this observation we start with the following definitions:

Definition 3.1 *Given a partial assignment $y \in \{0,1,*\}^n$ define a corresponding term and a corresponding clause*

- $t_y = \wedge_{i=1}^n x_i^{y_i}$
- $c_y = \overline{t_x} = \vee_{i=1}^n x_i^{1-y_i}$

where $x_i^0 = \overline{x_i}$, $x_i^1 = x_i$, $x_i^* = 1$, $x_i^{1-*} = 0$.

Similarly, given a clause c or term t, define corresponding partial assignments $y_c, y_t \in \{0,1,*\}^n$, such that:

- $(y_c)_j = \begin{cases} 1 & \textit{if } \overline{x_j} \in c \\ 0 & \textit{if } x_j \in c \\ * & \textit{otherwise} \end{cases}$

- $(y_t)_j = \begin{cases} 1 & \textit{if } x_j \in t \\ 0 & \textit{if } \overline{x_j} \in t \\ * & \textit{otherwise} \end{cases}$

where $(y_c)_j$, $(y_t)_j$ are the jth bits in y_c, y_t, respectively.

For example, if $y = (1 * 0)$, then $t_y = x_1 \wedge \overline{x_3}$, and $c_y = \overline{x_1} \vee x_3$. Clearly, $y_{c_y} = y$ and $c_{y_c} = c$ and similarly $y_{t_y} = y$ and $t_{y_t} = t$.

The following claim is immediate from the definitions:

Claim 3.2 *Let $y \in \{0, 1, *\}^n$ be a partial assignment, t a term, c a clause and $f : \{0,1\}^n \to \{0,1\}$. Then*
(1) $f_u(y) = 1$ if and only if $t_y \models f$.
(2) $t \models f$ if and only if $f_u(y_t) = 1$.
(3) $f_e(y) = 0$ if and only if $f \models c_y$.
(4) $f \models c$ if and only if $f_e(y_c) = 0$.

The above claim shows that unlike the pure Boolean case, where evaluating a function f on an assignment $x \in \{0,1\}^n$ is easy, in general, evaluating f on a partial assignment may be hard.

If we take the universal interpretation, the claim shows that evaluating $f_u(y)$ is equivalent to solving unsatisfiability for $(t_y \wedge \overline{f})$, and therefore is easy for f that is given in a CNF representation, but co-NP-Complete for f given in a DNF representation. On the other hand, if we take the existential interpretation, evaluating $f_e(y)$ is equivalent to solving satisfiability for $f \wedge \overline{c_y}$, and is therefore easy if f is given in a DNF representation, but when f is given in a CNF representation it is efficient only for some restricted subsets (e.g., Horn CNF formulas, 2-CNF formulas, $\log n$-clause CNF formulas). When using the abbreviated interpretation, evaluating f on a partial assignment is just as easy as evaluating it on a total assignment, the unique total assignment that is equivalent to this partial assignment.

Since the common assumption in knowledge representation and reasoning is that queries are represented in a CNF form, when trying to evaluate a query under the existential representation we need to make sure that we restrict ourselves to deal with queries that can be evaluated efficiently. We will show how to get around this problem by using a modified evaluation procedure.

As an aside we mention that the above discussion shows that when dealing with partial assignments, classification problems (e.g., evaluating a given function on an assignment) can be viewed as reasoning problems.

4 The Learning to Reason Framework

The Learning to Reason framework combines the interfaces to the world used by known learning models with the reasoning task and a performance criterion suitable for it. In the basic scenario studied in this framework, the intelligent agent is given access to its favorite learning interface, and is also given a grace period in which it can interact with this interface and construct a representation KB of the world W. The reasoning performance is measured only after this period, when the agent is presented with queries α from some query language, relevant to the world, and has to answer whether W implies α. We also study a scenario in which the agent learns and reasons in an on-line fashion. We first define the type of interactions available to the agent with the world under the partial observation assumption, and then define the corresponding Learning to Reason tasks.

4.1 The Interface

We define an interface to the world in the spirit of the known learning models (Valiant, 1984; Angluin, 1988), adapted to deal with partial assignments. As mentioned above we assume that one of the interpretations for partial assignments has been chosen, and is fixed for the duration of the learning scenario. When the interpretation is clear from the context we sometimes omit the subscript denoting which interpretation has been chosen. When we write $f(x)$, where f is a Boolean function and $x \in \{0,1,*\}^n$ is a partial assignment, we mean the value of f_e, f_u, or f_a on x, according to the interpretation chosen.

An interesting phenomenon occurs with oracles for partial assignments. Some of the oracles become "weaker" and some become "stronger" than their total assignments counterparts, in the sense that they supply less or more information, respectively, to the learner. This does not effect the plausibility of the oracles which should be evaluated with respect to the situations one is trying to model. We start by presenting the standard oracles:

The basic mode of interaction with the environment is via examples drawn according to a distribution that governs the occurrences of instances in the world. We assume here that the distribution is defined over $\{0,1,*\}^n$, the collection of all partial assignments.

Definition 4.1 *An* Example Oracle *for a function f, with respect to the probability distribution D over $\{0,1,*\}^n$, denoted $EX_D(f)$, is an oracle that when accessed, returns $(x, f(x))$, where x is drawn at random according to D.*

Definition 4.2 *A* Membership Query Oracle *for a function f, denoted $PMQ(f)$, is an oracle that when given an input $x \in \{0,1,*\}^n$ returns $f(x)$. We denote the standard membership oracle, which answers only on total vectors, by $MQ(f)$.*

Note that PMQ is a stronger oracle than MQ, since it answers all queries on total vectors and in addition all queries on partial vectors.

Definition 4.3 *An* Equivalence Query Oracle *for a function f, denoted $EQ(f)$, is an oracle that when given as input a function g, answers "Yes" if and only if $f \equiv g$. If it answers "No" it supplies a counterexample, namely, an $x \in \{0,1,*\}^n$ such that $f(x) \neq g(x)$. A counterexample x satisfying $f(x) = 1$ ($f(x) = 0$) is called a positive (negative) counterexample.*

The EQ oracle has more freedom than the EQ usually used when choosing the counterexamples (and is therefore weaker), since in addition to total assignments it can supply partial assignments.

The next oracles, introduced in (Frazier and Pitt, 1993), can be thought of as using the reasoning process itself as a source for examples.

Definition 4.4 *An* Entailment Membership Query Oracle *for a function f, denoted $EnMQ(f, \mathcal{Q})$, is an oracle that when given as input a function $g \in \mathcal{Q}$ answers "Yes" if $f \models g$ and "No" otherwise.*

Definition 4.5 *An* Entailment Equivalence Query Oracle *for a function f, denoted $EnEQ(f, \mathcal{Q})$, is an oracle that when given as input a function g, answers "No" if and only if there exists a counterexample in $\mathcal{Q}$, namely, a function $h \in \mathcal{Q}$ such that, either $f \not\models h$ but $g \models h$ (a negative counterexample), or $f \models h$ but $g \not\models h$ (a positive counterexample). When it answers "No" it also supplies the counterexample.*

We note that $EnEQ(f, \mathcal{Q})$ actually checks the equivalence of f and g relative to $\mathcal{Q}$. In fact, $EnEQ(f, \mathcal{Q})$ checks the equivalence of the least upper bounds of g and f in $\mathcal{Q}$. See (Selman and Kautz, 1996; Khardon and Roth, 1994b) for definitions and discussion of least upper bounds. Familiarity with this concept is not needed in the rest of this paper.

These are, of course, not all the oracles that can be defined, but rather those oracles that will be used in the rest of this paper. For example, in (Greiner and Schuurmans, 1992), an "entailment example oracle" is defined, which draws a query according to some probability distribution, and provides the right classification for it. In general, any reasonable way of interaction with the environment can be considered here.

In a particular learning scenario the agent may access a subset of the available oracles. We abstract this fact by defining the interface:

Definition 4.6 *We denote by $I(f)$ the* interface *available to the learner when learning f. This can be any subset of the oracles defined above, and might depend on some fixed but arbitrary and unknown distribution D over the instance space $\{0, 1, *\}^n$.*

The agent interacts with its environment while learning, via $I(f)$, and while reasoning, via the queries presented to it. The following class of queries plays an important role in our results:

Definition 4.7 *The class $\mathcal{Q}_C$ of* common queries *consists of Boolean functions with the following property: Every $\alpha \in \mathcal{Q}_C$ has a CNF representation, in which every clause is either (1) of size $\leq \log n$ or (2) a Horn-clause (contains at most one positive literal) or (3) a k-quasi-Horn clause (contains at most k positive literals).*

4.2 The Learning to Reason Task

As in the known learning models we distinguish between Learning to Reason in a "batch" type scenario, and "on-line" Learning to Reason.

To simplify notation, we assume from now on that all the functions discussed can be represented, in the corresponding representation class, with size polynomial in n (the number of variables), for some fixed polynomial.

In the batch scenario, the algorithm interacts with the environment, via $I(f)$, in order to acquire the knowledge for answering future queries. The performance of the algorithm is measured only after some "grace period", that must be of length polynomial in the size of the world description and in the quality of its performance, when it is required to decide, without further interaction with the world, if some query is entailed by f.

Definition 4.8 *An algorithm A is an* Exact Learn to Reason (E-L2R) *algorithm for the reasoning problem $(\mathcal{F}, \mathcal{Q})$, if there exists a polynomial $p()$ such that for all $f \in \mathcal{F}$, given access to $I(f)$, A runs in time $p(n)$ and then, when presented with any query $\alpha \in \mathcal{Q}$, A runs in time $p(n)$, does not access $I(f)$, and answers "Yes" if and only if $f \models \alpha$.*

When a probability distribution governs the occurrence of instances in the world we somewhat relax the requirements using the following restriction:

Definition 4.9 *The query α is called* (W, ϵ)-fair *if either $W \subseteq \alpha$ or $Prob_D[W \setminus \alpha] > \epsilon$.*

The intuition behind this definition is that in case that $W \not\models \alpha$ but the weight of W outside α is very small, we may allow the algorithm to err (and answer that $W \models \alpha$). When W and D are clear from the context we will say that α is ϵ-fair.

Definition 4.10 *An algorithm A is a* Probably Approximately Correct Learn to Reason (PAC-L2R) *algorithm for the reasoning problem $(\mathcal{F}, \mathcal{Q})$, if there exists a polynomial $p(,,)$ such that for any probability distribution D over $\{0,1,*\}^n$, for all $f \in \mathcal{F}$, on input ϵ, δ, and given access to $I(f)$, A runs in time $p(n, 1/\epsilon, 1/\delta)$ and then with probability at least $1-\delta$, when presented with any (f,ϵ)-fair query $\alpha \in \mathcal{Q}$, A runs in time $p(n, 1/\epsilon, 1/\delta)$, does not access $I(f)$, and answers "Yes" if and only if $f \models \alpha$.*

In the batch scenario above we did not allow access to $I(f)$ while in the query answering phase. In the on-line version however, we consider a query α presented to the algorithm as if presented by the oracle. Thus, a reasoning error may supply the algorithm a counterexample which in turn can be used to improve its future reasoning behavior. We allow the L2R algorithm to access $I(f)$ during this update, but *not* while answering a query. The following oracle is used to model the learning interface for the on-line scenario.

Definition 4.11 *A* Reasoning Query Oracle *for a function f and a propositional language $\mathcal{Q}$, denoted $RQ(f, \mathcal{Q})$, is an oracle that when accessed performs the following protocol with a learning agent A. (1) The oracle picks an arbitrary query $\alpha \in \mathcal{Q}$ and returns it to A. (2) The agent A answers "Yes" or "No" according to its belief with regard to the truth of the statement $f \models \alpha$. (3) If A's answer is correct then the oracle says "Correct". If the answer is wrong the oracle answers "Wrong". We call the oracle a* Reasoning Query Oracle with Counterexamples, *denoted $RQC(f, \mathcal{Q})$, if when $f \not\models \alpha$ and a reasoning mistake is made, it supplies a counterexample (i.e., $x \in f \setminus \alpha$ where $x \in \{0,1,*\}^n$).*

When learning, the algorithm is charged one mistake each time the reasoning query is answered incorrectly, and a successful learner should make a small number of mistakes.

Definition 4.12 *An algorithm A is a* Mistake Bound Learn to Reason (MB-L2R) *algorithm for the reasoning problem $(\mathcal{F}, \mathcal{Q})$, if A interacts with the reasoning oracle $RQ(f, \mathcal{Q})$, and there exists a polynomial $p()$ such that for all $f \in \mathcal{F}$, (1) A runs in time $p(n)$ (on each query) and answers "Yes" or "No" according to its belief with regard to the truth of the statement $f \models \alpha$, without accessing $I(f)$, (2) then runs in time $p(n)$ before it is ready for the next query (possibly, with accessing $I(f)$), and (3) for every (arbitrary infinite) sequence of queries, A makes no more than $p(n)$ mistakes.*

5 Reasoning with Partial Assignments

In this section we study knowledge representations that support efficient reasoning, and are suitable for an agent that interacts with its environment via partial assignments. As in the general case (when the interaction is via total assignments), formula-based representations do not support efficient reasoning. Thus, we resort to model based representations. However, given the type of interface we assume, it seems more plausible to search for a representation which consists of partial models. We start by discussing the notion of model based representations and presenting some of the technical details required. We then move on to define and study partial model based representations. Indeed, we show that it is unlikely that a simple partial assignment interaction can yield a total model based representation, justifying the use of partial model based representations.

5.1 Reasoning with Models

We have given a model-theoretic definition of the implication relation $\models$. The algorithm **MBR**, described in Figure 1 uses this definition in a straightforward way: it maintains a set of models

Algorithm MBR(Γ, α):

Test Set: A set $\Gamma \subseteq W$ of possible assignments.

Test: If there is an element $x \in \Gamma$ such that $\alpha(x) = 0$, return "No". Otherwise, return "Yes".

Figure 1: MBR: Model-Based Reasoning

$\Gamma \subseteq W \subseteq \{0,1\}^n$ as its knowledge base. To decide whether $W \models \alpha$ it checks, for all the models $z \in \Gamma$, whether $\alpha(z) = 1$. If for some z, $\alpha(z) = 0$, it says "No"; otherwise it says "Yes".

By definition, if $\Gamma = W$ this approach yields correct deduction, but representing W by explicitly holding *all* the possible models of W is not plausible. A model based approach becomes feasible if one can make correct inferences when working with a small subset of models. Some results on this line have been obtained (Kautz, Kearns, and Selman, 1995; Khardon and Roth, 1994b). In particular, previous work in (Khardon and Roth, 1994b) (using ideas from (Bshouty, 1995)) identified a small set of models, called the set of *characteristic models* of W, that supports correct reasoning. We briefly describe some of the relevant results we will need, culminating in Theorem 5.1 that identifies the models that are needed for the algorithm **MBR** to be correct and efficient.

Definition 5.1 (Order) *We denote by $\leq$ the* usual partial order *on $\{0,1\}^n$, the one induced by the order $0 < 1$. That is, for $x, y \in \{0,1\}^n$, $x \leq y$ if and only if $\forall i, x_i \leq y_i$. For an assignment $b \in \{0,1\}^n$ we define $x \leq_b y$ if and only if $x \oplus b \leq y \oplus b$, where $\oplus$ denotes the XOR operation (bitwise addition modulo 2). As with other order relations, $x \leq_b y$ can also be written as $y \geq_b x$, and if $x \leq_b y$ and $x \neq y$ we write $x <_b y$.*

Intuitively, if $b_i = 0$ then the order relation on the ith bit is the normal order; if $b_i = 1$, the order relation is reversed and we have that $1 <_{b_i} 0$.

Definition 5.2 *The* monotone extension of f *with respect to b is:*

$$\mathcal{M}_b(f) = \{x \mid x \geq_b z, \text{ for some } z \in f\}.$$

Definition 5.3 *A set B is a* basis for f *if $f = \bigwedge_{b \in B} \mathcal{M}_b(f)$. B is a basis for a class of functions $\mathcal{F}$ if it is a basis for all the functions in $\mathcal{F}$.*

It is known (Bshouty, 1995; Khardon and Roth, 1994b) that there are classes of functions which share the same small basis. In particular, the class of common queries defined above has a polynomial size basis B_c, and the set $B_{H_k} = \{u \in \{0,1\}^n \mid \text{weight}(u) \geq n-k\}$ is a basis for the class of k-quasi-Horn functions.

Definition 5.4 *Let $\mathcal{F}$ be a class of functions, and let B be a set of assignments in $\{0,1\}^n$. For $W \in \mathcal{F}$ we define the set $\Gamma = \Gamma_W^B$ of* characteristic models *to be the set of all minimal assignments of W with respect to the basis B. Formally,*

$$\Gamma_W^B = \cup_{b \in B}\{z \in min_b(W)\},$$

where

$$min_b(W) = \{z \mid z \in W, \quad \text{such that } \forall y \in W, z \not>_b y\}.$$

Algorithm Lazy-MBR(Γ, α):

Test Set: A set Γ of partial satisfying assignments.

Test: Given a CNF query α, if there is an element $x \in \Gamma$ which falsifies one of the clauses in α, deduce that $f \not\models \alpha$; Otherwise, $f \models \alpha$.

Figure 2: Lazy-MBR: Reasoning with Partial Models

Using these definitions we can identify the models that are needed for the algorithm **MBR**:

Theorem 5.1 (Khardon and Roth, 1994b) *Let $W \in \mathcal{F}$, $\alpha \in \mathcal{G}$ and let B be a basis for $\mathcal{G}$. Then $W \models \alpha$ if and only if for every $u \in \Gamma_W^B$, $\alpha(u) = 1$.*

5.2 Reasoning with Partial Models

We next consider whether partial assignments can be used in a similar way. We start by characterizing a useful set of partial models which is derived from partial views of characteristic models.

For any fixed k, there are $\binom{n}{k}$ subsets of size k of the n variables. Given an element $x \in \{0,1\}^n$ and a subset I of k variables, the projection of x on I is the partial model v defined by: $v_i = x_i$, for all $x_i \in I$, and $v_i = *$ otherwise. Let $\Gamma = \Gamma_f^B$ be the set of characteristic models, and fix B to be the basis for a class which includes k-CNF. (The basis B_{H_k}, of size $O(n^k)$ is sufficient, but there is a smaller basis, composed of a (n,k)-universal set (Naor and Naor, 1993).) Projecting all the elements of Γ on each of these subsets we get a set of $|\Gamma|\binom{n}{k}$ partial models.

Definition 5.5 *Let $\Gamma = \Gamma_f^B$ be the set of characteristic models. Then the set of all projections of elements of Γ on subsets of size k is denoted by $\Gamma_{f|k}^B$.*

Figure 2 describes the algorithm **Lazy-MBR** used for reasoning with partial models. The algorithm keeps partial assignments in its knowledge base, and assumes that the queries are given in CNF form. When it receives a CNF query α, the algorithm checks whether one of the partial assignments in its knowledge base falsifies one of the clauses in α. If it finds such a partial model, it says "No" and otherwise it says "Yes".

Note that the algorithm is slightly different from a normal model based algorithm **MBR**. The latter, when given a query and an assignment, will try to evaluate the query on this assignment to test whether the assignment satisfies the query. Since this may be hard for partial assignments, the lazy algorithm only tests for direct falsification of clauses in the query, and otherwise gives up. As the following theorem shows, if the knowledge base consists of the appropriate set of partial assignments, the lazy algorithm is guaranteed to succeed.

Theorem 5.2 *The algorithm* **Lazy-MBR**, *when using the set $\Gamma_{f|k}^B$, is correct for all queries $\alpha \in k$-CNF.*

Proof: Clearly, if $f \models \alpha$, model based reasoning answers correctly. Assume therefore that $f \not\models \alpha$. Theorem 5.1 implies that there exists a total model $z \in \Gamma_f^B$ such that $\alpha(z) = 0$. In particular, since α is a k-CNF, one of its clauses C_z must be falsified by z. That is $C_z(z) = 0$. Now consider the element z' which is the projection of z on the variables in C_z. Clearly z' is in $\Gamma_{f|k}^B$ and $C_z(z') = 0$. So the lazy model based algorithm will answer "No" due to z', and is therefore correct. ∎

	x_1x_2	x_1x_3	x_1x_4	x_2x_3	x_2x_4	x_3x_4
$\gamma_1 = 0000$	0 0	0 0	**00**	0 0	0 0	**00**
$\gamma_2 = 0101$	**01**	**00**	0 1	**10**	1 1	0 1
$\gamma_3 = 1110$	1 1	1 1	1 0	1 1	**10**	1 0
$\gamma_4 = 0100$	0 1	0 0	0 0	1 0	1 0	0 0

Figure 3: Projection Example

The previous theorem shows that given Γ_f^B we can generate a new model based representation, $\Gamma_{f|k}^B$, by projecting on all possible subsets of size k of the variables. It is interesting to ask whether the projection retains *all* of the information in Γ_f^B, so that one can reconstruct it from $\Gamma_{f|k}^B$. If true, this may enable us to answer a wider class of queries, in case Γ_f^B originally supported deduction also with queries from outside k-CNF, e.g., all common queries. The following claim shows this is not the case.

Claim 5.3 *The set Γ_f^B cannot be reconstructed from $\Gamma_{f|k}^B$. Furthermore, $\Gamma_{f|k}^B$ does not retain all the information needed in order to answer general queries supported by Γ_f^B.*

Proof: We prove the claim by exhibiting an example in which it is impossible to decide which of two sets of models created $\Gamma_{f|k}^B$. Let $n = 4$, $k = 2$, $\Gamma^{(3)} = \{\gamma_1, \gamma_2, \gamma_3\}$ and $\Gamma^{(4)} = \Gamma^{(3)} \cup \{\gamma_4\}$, where $\gamma_1 = 0000$, $\gamma_2 = 0101$, $\gamma_3 = 1110$ and $\gamma_4 = 0100$. Figure 3 shows the projections of elements in $\Gamma^{(4)}$ on subsets of size 2. The first three rows in the figure show all the projections of the elements in $\Gamma^{(3)}$. Since $\Gamma_{f|k}^B$ is a set, and all the elements in the forth line already appear in the previous lines, we get that it is impossible to decide whether $\Gamma_{f|k}^B$ is a projection of $\Gamma^{(3)}$ or $\Gamma^{(4)}$. However, these two sets yield different reasoning behavior, when the queries are not taken from 2-CNF.

For example, consider the functions $f = \{\gamma_1\} \cup \{\gamma_2\} \cup \{\gamma_3\}$ and $g = \{\gamma_1\} \cup \{\gamma_2\} \cup \{\gamma_3\} \cup \{\gamma_4\}$ whose model based representation is $\Gamma^{(3)}$ and $\Gamma^{(4)}$, respectively. Clearly, these functions respond differently to the reasoning query $\alpha = (x_1 \vee \overline{x_2} \vee x_3 \vee x_4) \in$ 4-CNF, but this query cannot be answered using $\Gamma_{f|k}^B$. This shows that $\Gamma_{f|2}^B$ does not support correct reasoning with queries in 4-CNF, and implies that the additional knowledge that $\Gamma^{(3)}$ is in fact a set consisting of characteristic models does not help the reconstruction.

■

Theorem 5.2 provides a polynomial procedure, as long as the original set Γ is of polynomial size and k is a constant. It would be desirable to relax the restriction on k, so that some dependency on n is possible. For example, when using $\Gamma_f^{B_c}$, Theorem 5.1 implies that we can reason correctly with $\log n$-CNF queries. However, the previous technique of projecting on all subsets does not work here since it creates too many partial assignments. The question is whether there is another way to project the assignments in Γ_f^B, so that a polynomial number of partial assignments is created, and the reasoning performance is correct.

Clearly, in order to answer all the queries that are disjunctions of size k we need to have, or be able to generate, all projections of the elements in Γ_f^B on subsets of this size. One such projection trivially exists, the projection over the set of all variables. We are looking, however, for sets of variables that are considerably smaller than n, since those are the elements we would like to keep in the representation. (Even though, we are willing to keep partial assignments that are longer than the queries, as long as they are not too long, and support correct reasoning.)

We call these sets of variables, on which we project the elements of Γ_f^B, *windows* of variables. Unfortunately, as we show below, small windows are too restrictive.

We say that a window r_1 covers window r_2 if $r_2 \subseteq r_1$. Notice that in this case, we can answer queries on variables from the window r_2 using the projection over r_1. Let R be a set of windows, and let L_k denote the set of $\binom{n}{k}$ windows of size k.

Claim 5.4 *If the largest window in R is smaller than $\sqrt{n}$ then either $|R| \geq n^{k/2}$ or R does not cover all the windows in L_k.*

Proof: If the largest window in R is of size m, then every window in R covers at most $\binom{m}{k}$ windows of size k. Since $m < \sqrt{n}$, the number of windows we need in order to cover all the $\binom{n}{k}$ windows of size k is at least:

$$N = \frac{\binom{n}{k}}{\binom{m}{k}} > \sqrt{n}^k = n^{k/2}.$$

∎

The last two claims suggest that model based reasoning with partial assignments is inherently limited. We can reason as long as the queries are in k-CNF for constant k but cannot go beyond that. While this might seem too strong a limitation, it may still explain some part of our common sense abilities where one is able to verify the validity of simple statements very well, but has difficulties with more elaborate statements.

On the positive side, we have shown in this section that even with the additional limitation of using only a partial model based representation, we can support more reasoning than when using the traditional formula based representation. Clearly, reasoning with respect to k-CNF queries is, in general, intractable, when using a CNF representation of f. In the next section we discuss the question of how to acquire model based representations.

6 L2R with Partial Assignments

In this section we present the Learning to Reason results of this paper. The results show that we can reason with respect to worlds for which the Learning to Classify problem is intractable and for which the traditional reasoning problem is intractable, exhibiting the utility of the Learning to Reason framework.

We present results that use various types of oracles, all of which provide only partial information about the environment. We assume throughout the section that one of the interpretations for partial assignments has been chosen, and will be fixed for the duration of the learning scenario.

6.1 A Sampling Approach

We first show that a simple sampling approach to reasoning, analyzed for total models in (Khardon and Roth, 1994a), works for partial assignments as well. The main difference is that, for partial assignments, the problem of model evaluation may be computationally hard (Claim 3.2). Therefore we have to restrict attention to queries that can be evaluated in polynomial time.

Let $\mathcal{Q}_E$ be a class of functions that can be evaluated in polynomial time on partial assignments. Notice that this class depends on the interpretation of the partial assignments. For the existential interpretation, this class includes all DNF formulas, Horn CNF formulas, 2-CNF formulas, and $\log n$-clause CNF formulas, but not general CNF representations.

The algorithm *Sample and Reason* first takes a sample of size $m = \frac{1}{\epsilon}(\ln |\mathcal{Q}_E| + \ln \frac{1}{\delta})$ examples from $EX_D(W)$. Let Γ be the set of positive examples sampled. Then, whenever presented with a query, the algorithm uses Γ and the algorithm **MBR** to answer the query, where model evaluation is done according to the appropriate interpretation.

A simple probabilistic argument yields the learning result. Let $\alpha \in \mathcal{Q}_E$ be an ϵ-fair query, and assume that $W \not\models \alpha$, then the probability that no example x in Γ is such that $\alpha(x) = 0$ is at most $(1-\epsilon)^m \leq e^{-(\ln |\mathcal{Q}_E| + \ln \frac{1}{\delta})} \leq \frac{\delta}{|\mathcal{Q}_E|}$. The probability that such an event happens for any query in $\mathcal{Q}_E$ is therefore at most δ. Observing that if $W \models \alpha$ a model based reasoning algorithm cannot make a mistake on α (since it cannot find a counterexample which does not exist) we get the following theorem:

Theorem 6.1 *The algorithm* Sample and Reason *is a PAC-L2R algorithm for the reasoning problem $(\mathcal{F}, \mathcal{Q}_E)$ for any class $\mathcal{F}$.*

The sampling result presented above may be used to motivate an approach that views the interaction with the agent's environment as an integral part of the reasoning process; with few assumptions and using the most basic interface between the agent and its environment, the L2R algorithm can support more than can be achieved by combining a traditional learning and reasoning algorithms. The result above implies that we can learn to reason for arbitrary world functions under the restriction that queries are ϵ-fair and can be evaluated efficiently. Later in this section we relax these requirements.

6.2 Relations Among Oracles

Before presenting more learning results we discuss the relations among several oracles that have been defined.

Lemma 6.2 *The oracle $PMQ(W)$ for the existential interpretation can be simulated efficiently by the oracle $EnMQ(W, disjunctions)$, and vice versa.*

Proof: The lemma follows from the relations pointed out in Claim 3.2. Given a partial assignment y presented to PMQ we present the clause c_y (Definition 3.1) to $EnMQ(W, disjunctions)$ and invert the (yes/no) answer received. The correctness follows from part (3) of Claim 3.2, namely, $W \models c_y$ iff $W_e(y) = 0$. On the other hand, given a query c presented to $EnMQ(W, disjunctions)$, we present the partial assignment y_c to PMQ and invert the answer. Correctness follows from part (4) of Claim 3.2, namely, that $W \models c$ iff $W_e(y_c) = 0$. Clearly, in both cases the simulation is efficient. ■

It is also easy to see that the oracles $RQ(W, Q)$, and $EnEQ(W, Q)$ are closely related. Namely RQ is, in some sense, an on-line version of $EnEQ$; if the learning algorithm has a hypothesis h that it presents to $EnEQ$ then, the counterexamples returned by $EnEQ$ are exactly those queries on which the algorithm will err, if presented by RQ. Thus, the information given to the learning algorithm is the same in both cases. A more subtle relation exists between $EnMQ$ and RQ in cases where the query class of RQ is restricted in a way that it must present the queries that the algorithm is interested in asking itself. In this case one can use RQ instead of $EnMQ$. We make use of all these relations in the following results.

6.3 L2R with k-CNF queries

While the class of k-CNF queries is expressive, the implication relation for this class can be captured by simple enumeration. Namely, it is sufficient to know the implication relation for disjunctions of size $\leq k$ and these can be easily combined to find the correct answer for any k-CNF expression. This follows from the fact that (for any W, c_1, c_2) if $W \not\models c_1 \wedge c_2$ then either $W \not\models c_1$ or $W \not\models c_2$. Therefore, given a CNF expression one can check the implication for every disjunction in it and answer "No" if one of these disjunctions is not implied by W. The crucial point, observed and utilized previously in (Moses and Tennenholtz, 1993), is that the number of disjunctions of size $\leq k$ is bounded by $3^k\binom{n}{k}$ and is therefore polynomial for constant k.

Using this observation we can present two L2R algorithms, that given access to $EnMQ(W, k\text{-}disjunctions)$ solve the reasoning problem $(\mathcal{F}, k\text{-CNF})$, for any class $\mathcal{F}$.

The first algorithm, *A-IMP* runs through the list of all k-disjunctions and uses $EnMQ(W, k\text{-}disjunctions)$ to collect a list of all k-disjunctions c such that $W \models c$. Then, given a query $\alpha = c_1 \wedge c_2 \ldots \wedge c_m$, the algorithm says "Yes" ($W \models \alpha$) if and only if *all* the disjunctions $c_1, c_2, \ldots, c_m$ in α are in this list. The second algorithm, *A-NIMP* also runs through the list of all k-disjunctions, but uses $EnMQ(W, k\text{-}disjunctions)$ to collect a list of all k-disjunctions c such that $W \not\models c$. Then, given a query $\alpha = c_1 \wedge c_2 \ldots \wedge c_m$, the algorithm says "No" ($W \not\models \alpha$) if and only if at least one of the disjunctions $c_1, c_2, \ldots, c_m$ in α are in its list. We have:

Claim 6.3 *Both A-IMP and A-NIMP are E-L2R algorithms for the reasoning problem $(\mathcal{F}, k\text{-}CNF)$ for any class $\mathcal{F}$. The number of queries the algorithms make is bounded by $3^k\binom{n}{k}$.*

For any W let $N(W)$ be the number of disjunction of size $\leq k$ which are not implied by W:

$$N(W) = \mid \{d \mid d \text{ has at most } k \text{ literals and } W \not\models d\} \mid .$$

Similarly, we denote by $P(W)$ the size of the complement set:

$$P(W) = \mid \{d \mid d \text{ has at most } k \text{ literals and } W \models d\} \mid .$$

Clearly, both $N(W)$ and $P(W)$ are smaller than the number of k-disjunctions.

The two algorithms presented above can be converted to on-line L2R algorithms, yielding a more natural view of the reasoning process as well as better bounds on the number of queries required. An on-line version for *A-IMP* proceeds as follows. Start with the empty list and initially predict "No". If a mistake is made (that is, $W \models \alpha$) add all the clauses in α to the list. Reasoning is done as above; say "Yes" iff all the clauses in α are in the list. Clearly, this algorithm never makes mistakes when $W \not\models \alpha$, and the number of mistakes it makes is bounded by $P(W)$.

Claim 6.4 *There exists a MB-L2R algorithm for the reasoning problem $(\mathcal{F}, k\text{-}CNF)$. The algorithm uses $RQ(W, k\text{-}CNF)$ and the number of mistakes it makes is bounded by $P(W)$.*

In order to understand the on-line version of *A-NIMP* it is useful to think of the problem in terms of concept learning. Consider the set of k-disjunctions as a new set of features $a_1, \ldots a_r$ ($r < 3^k\binom{n}{k}$). Notice that the concept of "not implied by W" is a disjunction over the a_i's, consisting of exactly those a_i's in the list maintained by *A-NIMP*. To see that, consider a query α as a list of its "active" (i.e., present) clauses. Queries that are not implied by W can be viewed as positive examples to the target concept, and queries that are implied by W as negative examples. The learning of this disjunction over the a_i's can be done in an on-line fashion, essentially using Blum's (1992) algorithm for monotone disjunctions. Start with the empty list and initially predict "No".

Algorithm A-OL-NIMP

1. Initialize $G = \emptyset$.
2. $\alpha \leftarrow RQ(W, Q)$
3. Use **Lazy-MBR**(G, α) to answer the query $W \models \alpha$.
4. If "wrong" then:

   if answered No then for all $z \in G$ and for all $c \in \alpha$ if z falsifies c then remove z from G.

   if answered Yes then for all $c \in \alpha$, compute the assignment $y_c \in \{0, 1, *\}^n$ (as in Definition 3.1), and if it was never added to G before then add y_c to G.
5. GoTo 2

Figure 4: The Algorithm *A-OL-NIMP*

When presented with a query α say "No" if and only if at least one of the clauses in α is in the list. If a mistake is made when $W \not\models \alpha$, add all the clauses a_i in α to the list. If a mistake is made when $W \models \alpha$, erase from the disjunction those a_i's (clauses) which appear in α. It is easy to see that this algorithm learns the required disjunction over the a_i's.

However, by viewing the knowledge representation maintained by the on-line algorithm as a disjunction over attributes (each corresponding to a clause), we lose some of the structure of the reasoning problem. To remedy that we now present a different on-line version of the algorithm *A-NIMP* that uses a model based representation as its knowledge representation, and reasons using the **Lazy-MBR** algorithm. In addition, the algorithm yields further savings in the number of queries used.

To understand the algorithm notice first that instead of keeping the k-disjunction c in the representation we can keep the partial assignment y_c. Verifying whether the k-disjunction c appears in the query α is equivalent to the lazy evaluation of α on y_c. When every clause is represented as a separate feature, and when c, c' are disjunctions (with no more than k literals) such that $c' \models c$, we still need to keep both. But, when using a model based representation it is sufficient to keep y_c since clearly also $c'(y_c) = 0$.

The algorithm *A-OL-NIMP*, described in Figure 4, maintains a model based representation G, initially empty, and uses the lazy evaluation algorithm **Lazy-MBR** (Figure 2) to respond to queries presented by RQ. Let $\alpha = c_1 \wedge c_2 \ldots \wedge c_m$ be the CNF representation of a query supplied by RQ, and assume that the algorithm makes a mistake on α. When $W \not\models \alpha$ and the algorithm responded "Yes", the algorithm produces a set of assignments and adds them to G. For each $c_i \in \alpha$, the algorithm produces the assignment y_{c_i} as in Definition 3.1. (For example, if $c = x_1 \vee \overline{x_3}$ then $y_c = (0 * 1)$.) The assignment is added to G only if it was never considered before. When $W \models \alpha$ and the algorithm responded "No", the algorithm removes from G the assignments that caused the mistake.

It is important to notice that, while the algorithm uses partial assignments, the following theorem is independent of the interpretation of the partial assignments. The reason is that it does not receive any examples from the oracles; those are produced internally by the algorithm. Namely, the oracle used, RQ, does not depend on the interpretation. A similar phenomenon occurs in Section 6.5, where entailment oracles are discussed.

Theorem 6.5 *For any class $\mathcal{F}$ of Boolean functions, the algorithm A-OL-NIMP, when given access to RQ(W, k-CNF), is a MB-L2R algorithm for the reasoning problem ($\mathcal{F}$, k-CNF). The number of mistakes A-OL-NIMP makes is bounded by min$\{m \cdot N(W), 3^k\binom{n}{k}\}$, where m is the maximal number of clauses in a query presented to the algorithm.*

Proof: It is sufficient to keep in G a (useful) counterexample for every disjunction c with up to k variables such that $W \not\models c$.

For every mistake the algorithm makes when $W \not\models \alpha$, there must be a $c \in \alpha$ such that $W \not\models c$. Whenever a mistake like this happens the algorithm adds an assignment y_c for every disjunction $c \in \alpha$. At least one of these assignments is a "good counterexample", in the sense that if falsifies the disjunction $c \in \alpha$ such that $W \not\models c$, and is thus useful for the **Lazy-MBR** algorithm. (Notice that by "falsifying a disjunction" we mean falsifying all the variables in the disjunction; this is obviously true by the construction.)

The way the assignments are generated guarantees that the algorithm will not make a mistake on any query which contains this clause, and since the algorithm uses **Lazy-MBR** for reasoning, correct counterexamples are never removed from G. We therefore get a bound of $N(W)$ for the number of mistakes in which the algorithm says "Yes" instead of "No". Every such mistake is responsible for producing at most $(m-1)$ mistakes of the other type. Since the algorithm introduces every clause at most once, we get the claimed mistake bound. ∎

Corollary 6.6 *For any class $\mathcal{F}$ of Boolean functions, there is a MB-L2R algorithm for the reasoning problem ($\mathcal{F}$, k-CNF) that uses RQ(W, k-CNF), and makes at most $2 *$ min$\{m \cdot N(W), P(W)\}$ mistakes, where m is the maximal number of clauses in a query presented to the algorithm.*

Proof: The algorithm simply alternates between the algorithm used in Theorem 6.5 and the algorithm used in Claim 6.4. When the currently used algorithm makes a mistake we update its representation and go on to use the other algorithm. This clearly results in the stated mistake bound. ∎

We note that for the universal and the abbreviated interpretations, it is possible to avoid the dependence on m in the above bound, by using *RQC* instead of *RQ*. For these interpretations the counterexample supplied by the oracle are useful[2] for model based reasoning.

The dependence of the mistake bound can be reduced in a different way, which holds for all the interpretations. This is done by converting the algorithm *A-OL-NIMP* to an attribute efficient algorithm using the Winnow algorithm (Littlestone, 1988), similar to what is done in (Blum, 1992). The resulting mistake bound is of the form $\log m \cdot N(W)$. The knowledge representation used by the new algorithm is not a list of models any more. Instead, a weighted sum of models is used for reasoning, where the weights are a function of the learning history.

To summarize, we mention again that the results of this section show that one can learn to reason even when the learning problem and the reasoning problem are hard. In particular, if W is an arbitrary Boolean circuit, the learning problem is known to be hard (independent of the representation (Kearns and Valiant, 1994)), and the reasoning problem is also hard, regardless of the class of queries (Cook, 1971). However, by restricting the queries to k-CNF we show that one can learn to reason.

[2]This does not hold for the existential interpretation since the counterexample can be adversarially chosen, such that **Lazy-MBR** will make repeated mistakes on the same query.

6.4 Using Stronger Oracles

The results of Section 5.2 suggest that the only way to improve the results we have shown so far, and reason with classes of queries that are wider than k-CNF, is to use total models in the representation. On the other hand, since the interface may supply short examples, and reconstruction is not possible, this seems to be impossible. In order to learn to reason with more expressive queries, we will have to use stronger oracles, which could be used to collect the set of (total) characteristic models Γ. Next we use a learning to reason result that was established in the context of total models. The result shows that the set of characteristic models can be learned and that this yields a learning to reason result. Recall that by $\mathcal{Q}_C$ we denote the class of all common queries.

Theorem 6.7 (Khardon and Roth, 1994a) *There is a MB-L2R algorithm, Ex-L2R-DNF, for the problem $(DNF, \mathcal{Q}_C)$. The algorithm interacts with the oracles $RQC(W, \mathcal{Q}_C)$ and $MQ(W)$ restricted to total models, maintains a model based representation $G \subseteq W$ (where W is the hidden world function), and when presented a query by RQC, it responds using G and the model based algorithm* **MBR**.

In the following we show that the same task can be performed using the appropriate partial assignments oracles.

Theorem 6.8 *There is a MB-L2R algorithm for the reasoning problem $(\mathcal{F}, \mathcal{Q}_C)$, for any class $\mathcal{F}$ and for all interpretations of partial assignments.*
(1) For the abbreviated and universal interpretations, the algorithm uses the oracles $RQC(W, \mathcal{Q}_C)$ and $MQ(W)$.
(2) For the existential interpretation the algorithm uses the oracles $RQ(W, \mathcal{Q}_C)$ and $PMQ(W)$.
The algorithm is polynomial in the DNF size of W and the size of the queries presented to it.

Proof: We show that the partial assignments oracles available here are as powerful as the total assignments oracles used in Theorem 6.7. We can therefore simulate algorithm *Ex-L2R-DNF* and hence learn to reason.

Clearly, a membership query oracle for partial assignments can answer all membership queries on total assignments. Instead of the oracle RQC for total assignments as in *Ex-L2R-DNF* we have access to either RQC with partial assignments in case (1) or to RQ in case (2). Therefore, we can use the given oracle for the purpose of interacting with the algorithm; what we need to do is to find a *total* counterexample to present to *Ex-L2R-DNF* when it makes a mistake.

Let α be a query on which the algorithm makes a mistake when interacting with RQ. First note that since the algorithm performs model based reasoning, and the models in its representation are true models of W, it must be the case that the algorithm said "Yes" whereas in fact $W \not\models \alpha$.

We argue by cases, according to the interpretation used. For the universal and abbreviated interpretations, we have access to RQC. That is, whenever a mistake is made, RQC returns a partial assignment $v \in W \setminus \alpha$ as a counterexample. In the abbreviated case, the unique 0-padded extension is the total counterexample needed. In the universal case, by definition, v must have an extension which falsifies a disjunction $d \in \alpha$. Such an extension is easy to find by finding a disjunction in which non of the literals is satisfied by v. (By definition, this extension satisfies W.)

For the existential interpretation, we have access to RQ, and therefore need to generate a total counterexample on our own. As in Theorem 6.5, we generate the counterexamples using the structure of the query α. Given $\alpha = \wedge c_i$, we generate the partial examples y_{c_i}. At least one of these examples, $v = y_{c_j}$, for some j, is positive for W. That is, the partial vector v is a counterexample. Next we show how, using PMQ, and these partial examples we generate a total counterexample.

First, using PMQ we find $v = y_{c_j}$ which is positive for W. Then, we extend v to a total vector which is still a counterexample. This can be done in a greedy manner, one bit at a time, using the oracle PMQ. Let v be the counterexample from above. Then, by definition, there exists some total example which is an extension of v and is positive for W. If there is a positive extension with 0 assigned to some bit, which is currently assigned $*$, (use a PMQ query to test this) then we can assign 0 to it, otherwise we can assign 1 to it. ∎

It is important to notice that while partial assignments were introduced to model situations in which the information available to the learner about the environment is limited, in the previous theorem we make use of fairly strong oracles. In particular, the partial membership query oracle PMQ is more powerful than the membership query oracle MQ. In the proof of the previous theorem it is used in order to complete partial assignments into total assignments. While completion of variables may be plausible in some situations, it may be more reasonable to limit the queries to a bounded number of specified literals. We leave this for future investigation.

As above, and similar to (Khardon and Roth, 1994a) this result shows that while learning DNF is still an open problem, it is possible to learn to reason with respect to DNF when the queries are common.

6.5 Using Entailment Queries

We next discuss learning to reason algorithms that use entailment oracles as the interface with the environment. Entailment oracles were introduced in Frazier and Pitt (1993) where an algorithm for learning to classify Horn expressions is developed. As argued in (Khardon and Roth, 1994a) this algorithm can be used as a learning to reason algorithm for the class of Horn queries and arbitrary W (and is polynomial for a certain class of functions discussed below).

When considering entailment oracles the issue of partial assignments in the interface does not exist. One way to enforce the interface to be partial is to assume that the equivalence entailment oracle returns as counterexamples only functions in, say k-Horn (i.e., Horn functions with up to k literals in a clause), and that the membership entailment oracle answers only when the function given to it is in this class. It is not hard to see that under this restriction the algorithm by Frazier and Pitt (1993) learns the k-Horn least upper bound of any theory, and therefore yields a Learning to Reason algorithm that can reason with any k-Horn query. However, we would not pursue this further here. Instead we show that if such restrictions are not made, then entailment oracles can be used to learn the set of characteristic models, and therefore learn to reason with a larger set of queries. However, as explained below, the two results are incomparable, since both the complexity of the algorithms and the strength of the oracles used are incomparable.

Theorem 6.9 *There is an E-L2R algorithm for the reasoning problem* $(\mathcal{F}, \mathcal{Q}_C)$*, for any class* $\mathcal{F}$*, when given access to* $EnEQ(W, \mathcal{Q}_C)$ *and* $EnMQ(W, disjunctions)$*. The algorithm is polynomial in* n*, the size of the DNF representation of* W *and the size of the queries presented to it.*

Proof: As in Theorem 6.8 the proof relies on simulating the oracles MQ and RQC used in Theorem 6.7.

First note that Lemma 6.2 implies that using $EnMQ(W, disjunctions)$ one can simulate $PMQ(W)$, in the case of the existential interpretation. In particular, one can simulate MQ using $EnMQ$.

We now show that $EnEQ$ can replace the oracle RQC. Recall that the algorithm maintains a set of models G, which it uses for model based reasoning. We will use the function $h = \wedge_{b \in B} \vee_{z \in G} \mathcal{M}_b(z)$ as the hypothesis of the algorithm. When the algorithm tries to access RQC, we instead call

$EnEQ(W, \mathcal{Q}_C)$ with the hypothesis h, and receive a counterexample $\alpha \in \mathcal{Q}_C$. We claim that only one type of counterexample can occur. Namely, it must be the case that $h \models \alpha$, and $W \not\models \alpha$. To prove this claim[3], assume that $h \not\models \alpha$. Then, by Theorem 5.1 there is a minimal model of h such that $\alpha(x) = 0$. Furthermore, x must be an element of G, since for all $b \in B$, $\exists z \in G$ such that $x \geq_b z$. But then since $G \subseteq W$ we also get $W \not\models \alpha$.

We present α that was received from $EnEQ$ to the algorithm, as if coming from RQC. Since the algorithm performs model based reasoning with G, by Theorem 5.1, it will make a mistake on α. We then compute a counterexample and return it to the algorithm, along with the "No" response. Computing the counterexample is done exactly in the same manner as in the proof of Theorem 6.8, where PMQ for the existential interpretation is used.

Finally, note that when the algorithm stops it responds correctly to all queries $\alpha \in \mathcal{Q}_C$. By the definition of $EnEQ$, when the algorithm stops (i.e., when $EnEQ$ returns "Yes"), $W \models \alpha$ iff $h \models \alpha$, for all $\alpha \in \mathcal{Q}_C$. Therefore, by Theorem 5.1, model based reasoning with G is correct.

■

We note that the strength of the oracles used is incomparable with those used by Frazier and Pitt (1993). When using $EnMQ$, we used a "stronger" version of it, since we allowed ourselves to ask queries that are arbitrary disjunctions, compared to Horn disjunctions used there. On the other hand we used a "weaker" version of $EnEQ$ since it was allowed to return counterexamples in $\mathcal{Q}_C$, compared with Horn disjunctions used there. The results are also incomparable in complexity since (as argued in (Khardon and Roth, 1994b)) the algorithm in Frazier and Pitt (1993) requires time which is polynomial in the propositional Horn representation of the least upper bound of W, whereas Theorem 6.9 requires time polynomial in the DNF size of W, and the two are in general incomparable. Another important difference relates to partial assignments. As discussed in the previous section the oracles used here, PMQ and also $EnEQ$ can give more information than their total-assignment counterparts. This is different from the algorithm of Frazier and Pitt (1993) which always asks $EnMQ$ queries which are shorter than the disjunctions it receives as counterexamples.

7 Conclusions

Most of the work on learning and reasoning has considered those as separate tasks, assuming that eventually the tasks will be combined to yield intelligent behavior. However, in their current formalization, both learning and reasoning problems are known to be computationally hard. Moreover, it may not be possible to use the output of the learning stage for reasoning in a straightforward way.

The *Learning to Reason* framework emphasizes the role of inductive learning in achieving efficient reasoning, and the importance of studying reasoning and learning phenomena together.

In this paper we considered the problem of Learning to Reason when the interaction with the world supplies only partial information. We have shown, in the context of partially observable worlds, that the agent truly gains additional computational power in the Learning to Reason framework. Namely, one can learn to reason even when the separately defined tasks of learning and reasoning are hard.

In the problem considered in this paper the agent interacts with its environment using examples that are only partially specified. Its performance is measured relative to the task of logical deduction. Namely, it has to determine whether a given query, supposed to capture the situation at hand, is true in the world.

[3]This can be traced to the fact that $h \models W^B_{lub}$, where W^B_{lub} is the least upper bound of W in $\mathcal{Q}_C$ (Khardon and Roth, 1994b). We give an argument from first principles to avoid using the notion of least upper bounds.

Several natural interpretations of partial assignments were considered, and it was shown that logical deduction retains the same meaning with respect to those. The Learning to Reason tasks were studied with respect to these interpretations.

Several of our results are based on a model based approach to reasoning. We have shown that a knowledge base which consists of a collection of partial examples can be used to support correct and efficient reasoning with respect to the world, and a restricted class of queries. Furthermore, such representations can be learned, yielding a learning to reason algorithm.

We studied several interfaces of the agent with its environment that are suitable to handle partial assignments, and have proved learning to reason results whose strength depend on the type of interaction assumed. In particular, random partial assignments can be used to reason with respect to ϵ-fair queries, the oracle RQ restricted to k-CNF queries can be used to reason with respect to the same class of queries, and stronger oracles can be used to support reasoning with respect to all common queries.

This work highlights the fact that the study of learning can be incorporated within the study of other high level cognitive tasks and can contribute to the understanding of those tasks and motivates the study of learning problems that arise in this way.

Acknowledgments

We are grateful to Moti Frances, Wolfgang Maass, Les Valiant and anonymous referees for their useful comments on earlier versions of this paper.

References

Angluin, D. 1988. Queries and concept learning. *Machine Learning*, 2(4):319–342, April.

Ben-David, S. and E. Dichterman. 1993. Learning with restricted focus of attention. In *Proc. 6th Annu. Workshop on Comput. Learning Theory*, pages 287–296. ACM Press, New York, NY.

Blum, A. 1992. Learning Boolean functions in an infinite attribute space. *Machine Learning*, 9(4):373–386, October.

Blum, A., L. Hellerstein, and N. Littlestone. 1991. Learning in the presence of finitely or infinitely many irrelevant attribute. In *Proceedings of COLT '91*.

Bshouty, N. H. 1995. Exact learning via the monotone theory. *Information and Computation*, 123(1):146–153.

Cadoli, M. 1995. *Tractable Reasoning in Artificial Intelligence*. Springer-verlag. Lecture notes in Artificial Intelligence, vol. 941.

Cook, S. A. 1971. The complexity of theorem proving procedures. In *3rd annual ACM Symposium of the Theory of Computing*, pages 151–158.

Doyle, J. and R. Patil. 1991. Two theses of knowledge representation: language restrictions, taxonomic classification, and the utility of representation services. *Artificial Intelligence*, 48:261–297.

Druker, H., R. Schapire, and P. Simard. 1993. Improving performance in neural networks using a boosting algorithm. In *Neural Information Processing Systems 5*, pages 42–49. Morgan Kaufmann.

Fagin, R., J. Halpern, and M. Vardi. 1995. A nonstandard approach to the logical omniscience problem. *Artificial Intelligence*, 79:203–240.

Frazier, M. and L. Pitt. 1993. Learning from entailment: An application to propositional Horn sentences. In *Proceedings of the International Conference on Machine Learning*. Morgan Kaufmann.

Greiner, R., A. Grove, and A. Kogan. 1996. Exploiting the omission of irrelevant data. In *Proceedings of the International Conference on Machine Learning*, pages 216–224.

Greiner, R., A. Grove, and D. Roth. 1996. Learning active classifiers. In *Proceedings of the International Conference on Machine Learning*, pages 207–215.

Greiner, R. and D. Schuurmans. 1992. Learning useful Horn approximations. In *Proceedings of the International Conference on the Principles of Knowledge Representation and Reasoning*, pages 383–392.

Hancock, T., R. Greiner, and R. B. Rao. 1994. Exploiting the absence of irrelevant information. In *AAAI Fall Symposium on Relevance*, pages 178–183.

Kautz, H., M. Kearns, and B. Selman. 1995. Horn approximations of empirical data. *Artificial Intelligence*, 74:129–145.

Kearns, M.J. and L.G. Valiant. 1994. Cryptographic limitations on learning Boolean formulae and finite automata. *Journal of the ACM*, 41(1):67–95.

Khardon, R. 1996. Learning to take actions. In *Proceedings of the National Conference on Artificial Intelligence*, pages 787–792.

Khardon, R. and D. Roth. 1994a. Learning to reason. In *Proceedings of the National Conference on Artificial Intelligence*, pages 682–687.

Khardon, R. and D. Roth. 1994b. Reasoning with models. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1148–1153. To appear in Artificial Intelligence Journal.

Khardon, R. and D. Roth. 1995. Default-reasoning with models. In *Proceedings of the International Joint Conference of Artificial Intelligence*, pages 319–325, August. To appear in Artificial Intelligence Journal.

Kirsh, D. 1991. Foundations of AI: the big issues. *Artificial Intelligence*, 47:3–30.

Levesque, H. 1984. A logic of implicit and explicit belief. In *Proceedings of the National Conference on Artificial Intelligence*, pages 198–202.

Levesque, H. 1992. Is reasoning too hard ? In *Proceeding of the 3rd NEC research Symposium.*

Levesque, H. and R. Brachman. 1985. A fundamental tradeoff in knowledge representation and reasoning. In R. Brachman and H. Levesque, editors, *Readings in Knowledge Representation.* Morgan Kaufman.

Littlestone, N. 1988. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318.

McCarthy, J. 1958. Programs with common sense. In *Proceedings of the Symposium on the Mechanization of Thought Processes*, volume 1, pages 77–84. National Physical Laboratory. Reprinted in Minsky's (ed.) *Semantic Information Processing*, MIT Press(1968), 403–409. Also in R. Brachman and H. Levesque, Readings in Knowledge Representation, 1985.

Moses, Y. and M. Tennenholtz. 1993. Off-line reasoning for on-line efficiency. In *Proceedings of the International Joint Conference of Artificial Intelligence*, pages 490–495, August.

Naor, J. and M. Naor. 1993. Small-bias probability spaces: Efficient constructions and applications. *SIAM Journal on Computing*, 22(4):838–856, August.

Nilsson, N. J. 1991. Logic and artificial intelligence. *Artificial Intelligence*, 47:31–56.

Papadimitriou, C. H. 1991. On selecting a satisfying truth assignment. In *Proc. 32nd Ann. IEEE Symp. on Foundations of Computer Science*, pages 163–169.

Reiter, R. 1987. Nonmonotonic reasoning. In *Annual Reviews of Computer Science.* pages 147–188.

Roth, D. 1995. Learning to reason: The non-monotonic case. In *Proceedings of the International Joint Conference of Artificial Intelligence*, pages 1178–1184, August.

Roth, D. 1996. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1-2):273–302, April.

Schuurmans, D. and R. Greiner. 1994. Learning default concepts. In *Proceedings of the Tenth Canadian Conference on Artificial Intelligence (CSCSI-94).*

Selman, B. 1990. *Tractable Default Reasoning.* Ph.D. thesis, Department of Computer Science, University of Toronto.

Selman, B. and H. Kautz. 1996. Knowledge compilation and theory approximation. *Journal of the ACM*, 43(2):193–224, March.

Shastri, L. 1993. A computational model of tractable reasoning - taking inspiration from cognition. In *Proceedings of the International Joint Conference of Artificial Intelligence*, pages 202–207, August.

Valiant, L. G. 1984. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, November.

Valiant, L. G. 1995. Rationality. In *Workshop on Computational Learning Theory*, pages 3–14, July.